Rec'd PCT/PTO 18 JAN 2005
10/521581
WO 2004/008698 PCT/CA2003/001043

APPARATUS, SYSTEM AND METHOD FOR THE TRANSMISSION OF DATA WITH DIFFERENT QOS ATTRIBUTES

FIELD OF THE INVENTION

The present invention relates to an apparatus, system and method for transmitting data flows that have different quality of service (QoS) attributes over a network link. More specifically, the present invention relates to an apparatus, system and method for providing and managing quality of service (QoS) for data flows transmitted over at least one link in a data network capable of transmitting data with different QoS requirements and/or attributes.

BACKGROUND OF THE INVENTION

One of the most ubiquitous data networks to date has been the Internet which is a packet data network employing the Internet Protocol (IP) as its network layer protocol. IP provides many advantages as a network layer protocol, including robustness and simplicity of implementation and one of its original principles is that the network layer need not know anything about the contents of the packets it transmits. In other words, to IP, a packet was a packet was a packet. Similarly, IP is intended to work independently of the physical layer of the network. In other words, IP is ignorant of whether the network it is operating over is an optical network or a wireline network, etc.

More recently, much effort has been spent in creating converged networks wherein diverse types of data can be handled by a single network. For example, packet data networks are now often able to carry voice (telephony) data, "pure" data (such as data file transfers, emails, etc.), fax data, streaming video, video conferencing, etc. Many of these converged networks also employ IP as their network protocol.

While IP continues to be the network layer protocol of choice for many modern networks, data from different user applications in a converged network can require different transmission characteristics to be provided for them. For example, data packets from and/or to telephony voice coders require relatively low end-to-end transmission latencies, but can accommodate relatively high error rates and/or dropped packets. In contrast, data packets from a file transfer protocol (FTP) session can accommodate relatively long latencies, but cannot well accommodate errors or dropped packets. Data packets carrying fax data using the T.38 protocol

require both low latency and low error rates. The specific requirements for the acceptable transmission of the contents of a data packet are generally referred to as the required quality of service (QoS) for the data. As originally designed, IP did not contemplate providing different QoS levels for packets.

To provide some measure of QoS control in IP, the fourth version of the protocol, typically referred to as IPv4, provided a Type of Service (TOS) byte in the standard IP header, as defined in IETF RFC 791 (1981) (available from www.ietf.org). The first three bits (0 to 2) of the TOS byte represent a precedence field with eight defined values (specifically, "Network Control", "Internetwork Control", "CRITIC/ECP", "Flash Override", "Flash", "Immediate", "Priority" and "Routine"). Apart from Router Table updates and similar networking functions, the precedence bits are not widely used in most networks today.

The next four bits (bits 3 to 6 - typically referred to as the TOS bits) of the TOS byte represent flags indicating a desired type of service and the last bit (bit 7) is left blank. The TOS bits essentially act as flags to request from the network service levels to: minimize delay; maximize throughput; maximize reliability; or minimize monetary cost. An application can set any one of the four TOS bits as desired or, if none are set, it is assumed that normal service is desired. Suggested settings of the TOS bits for common applications were described in IETF RFC 1600 (Reynolds and Postel, 1992) and IETF RFC 1609 (Almquist, 1992). As specific examples, it is recommended in these documents that FTP control packets have the minimize delay bit set, that FTP data packets have the maximize throughput bit set and that usenet (NNTP) packets have the minimize monetary cost bit set.

While TOS provides some ability to provide QoS, it is very limited. Accordingly, many other attempts have been made to provide QoS mechanisms for IP. For example, RSVP has been proposed as a mechanism for providing QoS assurance in an IP network. Essentially, RSVP reserves resources from network components to provide virtual connections through the otherwise connectionless IP network. RSVP suffers from difficulties in that it is not yet broadly supported and that, even when supported, it assumes that sufficiently large amounts of network resources (bandwidth, etc.) are available to permit some of these resources to be reserved for specific users and/or applications and it can result in inefficient use of these resources.

Additional problems exist when trying to provide QoS over an IP network that is implemented on a physical layer with higher error probabilities. Specifically, IP was designed

for, and assumes, a reliable physical layer, such as wired Ethernet or the like where congestion may be a problem, but where large amounts of generally reliable bandwidth are available. To date, implementing QoS-enabled IP based networks over less reliable physical layers, such as radio channels, has been difficult.

Further, to date most attempts at implementing QoS for IP networks have taken an end to end approach and have not addressed networks with heterogeneous physical layers, such as networks with both wired and radio links. Different physical layers can result in much different QoS mechanisms being required.

Also, in radio systems, such as that proposed by the third generation partnership project (3GPP), QoS must typically be provided at the physical layer of the system, requiring different channels to provide different QoS levels. Obviously, this can severely limit the range of QoS offerings that can be provided in a network.

It is desired to have a broadly QoS-enabled IP network which can operate on a variety of physical layers, including relatively unreliable layers such as radio channels, and/or in networks with heterogeneous physical links.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a novel apparatus, system and method for transmitting at least two data flows over a telecommunications link structured into at least two channels, each data flow having different QoS attributes which obviates or mitigates at least one of the above-identified disadvantages of the prior art.

According to a first aspect of the present invention, there is provided a method of transmitting at least two data flows over a telecommunications link, wherein each data flow can have a different set of quality of service attributes defined for it, comprising the steps of:

(i) receiving a packet for transmission over said link;

(ii) examining said packet to determine an appropriate set of quality of service attributes required for it;

(iii) placing said examined packet into one of a plurality of logical channel queues, said one logical channel queue having defined therefor quality of service attributes corresponding to the determined quality of service attributes required for said packet;

(iv) determining a data transmission capacity for said link and selecting one of said

plurality of logical channel queues which holds data to be transmitted with the highest priority for transmission and packaging and transmitting as much data from said logical channel queue as can be packaged to fit within said determined transmission capacity of said channel; and

(v) repeating steps (i) through (iii) and step (iv) as necessary.

Preferably, the telecommunications link is structured into two or more channels, each of which can have different amounts of data transmission capacity and each of the logical channel queues can be assigned to one or more of the channels. In this case, the determined data transmission capacity is determined for each channel and, for each channel, one of the assigned logical channel queues is selected for transmission. Also preferably, each of the channels can be either a dedicated channel, between a pair of endpoints, or a broadcast channel between an endpoint and a plurality of other endpoints. Also preferably, for one to many endpoint configurations, a set of logical channel queues is defined at the one endpoint for transmissions to each of the many endpoints.

According to another aspect of the present invention, there is provided a system for the transmission of data flows from a first endpoint to one or more of a plurality of other endpoints through a link where each flow can have different quality of service attributes defined therefor, comprising:

a network interface at said first endpoint to receive packets from said data flows for transmission through said link to said other endpoints;

a set of logical channel queues, each of said logical channel queues in said set being operable to queue a received packet and having a set of defined quality of service attributes defined for the contents of the queue;

a packet classifier to examine said received packets to determine the quality of service attributes for said packet and to place received packets into selected ones of said logical channel queues with corresponding quality of service attributes; and

a link controller which determines the available data transmission capacity to said plurality of end points and which selects for transmission a portion of a packet from the logical channel queue whose contents have the highest priority, the link controller segmenting the packet as necessary to have the portion fit within the data transmission capacity of said link.

Preferably, the system maintains a set of logical channel queues at the first end point for each of said plurality of other endpoints. Also preferably, the link is structured into two or more

channels and the link controller selects, for each of the at least two channels, a portion of a packet from a logical channel queue for transmission to at least one of the plurality of endpoints. Also preferably, at least one of these least two channels is a dedicated channel between the first end point and one of the plurality of endpoints and another of the at least two channels is a broadcast channel from the first endpoint to at least two or more of the plurality of other endpoints. Also preferably, the link controller is operable to change the structure of each dedicated channel to alter its data transmission capacity to meet the needs of the endpoints.

The present invention provides for the efficient utilization of a shared resource, such as a radio link, in a network including heterogeneous links. Data is organized in flows and each flow can be provided with its own logical channel with its own set of QoS attributes. These attributes can include prioritization, latency restrictions, data rate requirements, reliability requirements, etc. Data flows with wildly different quality of service attributes can be multiplexed onto a single link, which can have variable data transmission capacities, and the necessary differentiated quality of service for these flows can be efficiently provided. The range and/or types of attributes are not particularly limited and can include attributes such as: whether ARQ is to be provided; whether header compression is to be performed; priority; tolerance or intolerance to latency; etc.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the present invention will now be described, by way of example only, with reference to the attached Figures, wherein:

Figure 1 shows a schematic representation of a network and connected end users, the network being heterogeneous and comprising wireline and radio telecommunicatiosn links;

Figure 2 shows end user terminals connected to a schematic representation of a CPE;

Figure 3 shows a schematic representation of a QoS processor at a base station, including multiple prioritization and queuing engines;

Figure 4 is a flowchart representing the steps performed within the QoS processor upon receipt of a packet for transmission;

Figure 5 is a flowchart representing the steps performed by a Radio Link Controller to select and transmit a segment of a packet;

Figure 6 shows a plot of transmitted packet segments of different priorities;

Figure 7 shows a reassembly controller; and

Figure 8 shows a schematic representation of a prioritization and queuing engine at a CPE.

DETAILED DESCRIPTION OF THE INVENTION

A telecommunication system 20 is illustrated in Figure 1 and comprises a variety of end user data 24 and telephony 28 terminals connected by a network 32. Network 32 can be any suitable telecommunications network, such as a packet network implemented using IP and running on a wireline or optical backbone, etc.

As illustrated, some terminals can be connected to network 32 via wired links 36 such as, for example, T1, xDSL, cable modem, dial up, fiber optic systems, etc. Other terminals can be connected via wireless links 40 that extend between radio base stations 44 and subscriber stations, such as customer premises equipment (CPE) 48, to which the terminals are connected. The base station 44 and subscriber stations can be components of any radio transmission system suitable for the transmission of data and, in a present embodiment, is the AMOSPHERE™ system manufactured and sold by the assignee of the present invention. The AMOSPHERE system employs wideband CDMA between NPM base stations and SOMAport™ CPEs and provides for the fully converged transport of data from CPEs 48, over a shared wireless link 40, to base stations 44 and to and from core network 32 via a backhaul 52. Backhaul 52 can be any suitable backhaul link including, but not limited to, T3, OC3, microwave or other telecommunications links.

Figure 2 shows a schematic representation of one possible embodiment of CPE 48 in system 20. As shown, CPE 48 includes a microprocessor assembly 50 that includes one or more input/output ports allowing data terminals 24 to be connected to CPE 48. Similarly, CPE 48 includes a subscriber line interface circuit (SLIC) assembly 54 that provides one or more standard RJ-11 ports, or other suitable interfaces, to connect one or more telephony devices 28 to CPE 48. SLIC assembly 54 also includes A/D and D/A converters, as well as any desired voice codecs, to connect telephony devices 28 to microprocessor assembly 50. CPE 48 can also act as a wireless access point for wireless communication to and from end user terminals within the customer's premises, via any appropriate technology including, but not limited to, 802.11a, 802.11b or 802.11g radio access points, a Bluetooth transceiver, infrared transceiver, or any

other suitable technology as will occur to those of skill in the art. Other embodiments of CPE 48 will occur to those of skill in the art, including PCMCIA or other form factor devices that can be connected, or installed in, various user terminals and which can operate in nomadic or mobile applications.

As shown, microprocessor assembly 50 connects, via a modem 60, to a radio transceiver 64 which is, in turn, connected to antenna 68. Data is received over radio link 40 from a base station 44, via antenna 68, and is amplified by radio 64 and demodulated/decoded by modem 60 which provides the resulting data to microprocessor assembly 50. Microprocessor assembly 50 either acts on the received data, if it is a control signal, or passes the data, in the appropriate format, to the appropriate data terminal 24 or to telephony device 28 via SLIC assembly 54.

Similarly, data is received by microprocessor assembly 50 from a data terminal 24 or telephony terminal 28, via SLIC assembly 54, and is modulated and encoded by modem 60, amplified by radio 64 and transmitted, via antenna 68, over radio link 40 to base station 44.

As will be apparent, radio link 40 is subject to a variety of conditions and/or restrictions. For example, typically only a limited amount of radio spectrum (bandwidth) is available for use by radio link 40 and this bandwidth must be shared between all CPEs 48. Also, typically the total amount of data capacity available from the base station 44 to individual CPEs 48 (the downlink) is significantly higher than the data capacity available from CPEs 48 to base station 44 (the uplink). For example, an aggregate rate of twelve megabits per second (mbps) may be available in the downlink while the uplink may be limited to one mbps, or less.

Further, the radio path characteristics between individual CPEs 48 and a base station 44 will vary widely due both to radio propagation factors (distance, orientation, etc.) and due to variations with time (fading, localized interference sources, etc.). Thus, any given CPE 48 will experience radio path characteristics at some times which allow it to receive or send data at some maximum rate (e.g. - five megabits per second) and will experience radio path characteristics at other times which only allow it to receive or send data at some minimum rate (e.g. - five hundred thousand bits per second) and each CPE 48 will experience radio path characteristics between each of these extremes at other times. Accordingly, system 20 must be able effectively use the capacity of radio link 40 even though that capacity will change with time and, in particular, the capacity between individual CPEs 48 and base station 44 can experience significant changes over time.

Also, as radio channels are generally more prone to transmission errors than some other physical media, such as wireline links, it is necessary to appropriately format data for transmission over radio channels, hereinafter referred to as "packaging" of the data. This packaging can include employing error-correcting codes, interleaving the data, selecting the modulation employed, etc. One of the consequences of this packaging is that, typically, the maximum size of the physical layer transport block (i.e. – the largest amount of data that can be transmitted by the physical layer in a time period, such as a single frame) of a radio channel is smaller than would be the case for wireline transmissions. For example, in a wireline network the transport block for an Ethernet system can be fifteen hundred bytes or more in size, while in the above-mentioned AMOSPHERE system, the transport block may be only eleven bytes at minimum on the uplink (CPE 48 to base station 44) and nineteen bytes at minimum on the downlink (base station 44 to CPE 48). Also, rather than transmitting large packets which may have a greater probability of encountering a transmission error, it can be advantageous to transmit smaller packets which can have a reduced probability of experiencing an error and which, in the event of an error, can be retransmitted in less bandwidth than a larger packet would require. Thus, packets from a wireline system and/or a user terminal will usually require fragmentation prior to transmission over radio link 40.

While fragmentation is employed in conventional IP systems when necessary, reassembly of fragmented packets is not typically performed until the packets arrive at their final destination. Further, the overhead (headers, etc.) in performing fragmentation can make it very expensive, in terms of efficient use of bandwidth and resources, to implement even when it is required. As will be described in detail below, in the present invention little overhead is required for segmentation of packets and reassembly of segmented packets to the pre-segmentation transport block size can be performed when the segments are received at the end of radio link 40.

Thus, effective use of the radio link 40 requires management of a limited, shared, resource where the capabilities of individual terminals to use that resource can vary widely over time and where the fragmentation of data packets will be the norm. On top of these issues, if QoS is required for the data an extra degree of complexity must be included.

Figure 3 shows the QoS processor100 that runs at base stations 44 in system 20. A separate QoS processor 100 is available for each shared radio link 40, which in the above-mentioned AMOSPHERE system means that each radio sector in a multi-sector base station 44

has its own QoS processor 100. QoS processor 100 includes a network interface port 104_x for each CPE 48_x it serves, through which information to be transmitted to the respective CPE 48_x is delivered. Each network interface port 104 feeds into a respective prioritization and queuing engine (PQE) 108_x, each of which in turn includes a data classifier 112 and a set of logical channel queues LC_0 to LC_i. In the illustrated embodiment of the invention, each PQE 108 includes sixteen logical channel queues, specifically, LC_0 through LC_{15}. Logical channel queues LC_i comprise queues of packets to be transmitted, where each entry in a queue holds one packet. The size of the packets placed into the queues of different logical channel queues LC_i can differ as necessary. For example, a voice telephony data packet may be twenty bytes while a web browsing session data packet can be several hundred bytes or more.

As each packet arrives at a network interface port 104_x, classifier 112_x examines the packet to route it to an appropriate one of the up to sixteen logical channel queues LC_i in PQE 108_x. Classifier 112_x can perform the classification based upon a variety of factors, including: the IP header TOS field contents; the source and/or destination IP addresses and ports; the payload type (if known); payload length; etc.

Typically, each data flow from a base station 44 to a CPE 48, or vice versa, is assigned to a different logical channel queue LC_i and each logical channel queue LC_i has an appropriate priority and set of QoS attributes defined for it. A data flow can be any communication need served by the network, for example a web browsing session can be one flow, while each of two telephony calls can be other flows and a file transfer can be a fourth flow.

As used herein, and as discussed further below, the term "QoS attributes" can include a wide variety of attributes including., but not limited to: traffic shaping; segmentation prohibition; priority; data rate; latency; reliability; header compression techniques, probability of block errors; etc.

For example: LC_0 and LC_1 can be defined with the low latency, high error tolerance, header compression QoS attributes suitable for voice data, if two voice connections are being provided between the base station 44 and a CPE 48; LC_2 can be defined with the moderate tolerance for latency, low error tolerance QoS attributes appropriate to a web browsing session; LC_3 can be defined with suitable QoS attributes (tolerance for high latency and low tolerance for errors, i.e. – ARQ enabled) for email; LC_4 can be defined with appropriate QoS attributes for streaming media, such as video; LC_5 can be defined with QoS attributes suitable for system

control and messaging signals; and LC_6 can be defined with the low latency, low error tolerance QoS attributes suitable for fax data using the T.38 protocol, etc. Logical channel queues can be instantiated as needed and the QoS attributes can be defined and/or modified as required, as discussed further below.

Figure 4 shows a flowchart of the above-described process. At step 120, a packet is received at a PQE 108 from network interface port 104 and is classified by classifier 112. Classifier 112 determines the appropriate QoS attributes for the received packet. At step 124, classifier 112 determines if a logical channel queue LC is available for the packet and if such a logical channel queue is not available, the method creates the required logical channel queue at step 128, either by creating a new logical channel queue with the necessary QoS attributes, or by modifying the QoS attributes of an existing, empty, logical channel queue. When the required logical channel queue is available, the method enqueues the received packet at step 132.

As described below, each logical channel queue LC_i performs prioritized scheduling of packets enqueued in it and data is transmitted from the logical channel queue LC_i in a non-preemptive manner. Specifically, as each packet is added to a logical channel queue, its priority, relative to the already enqueued packets, is determined. The highest priority packet in the queue is selected for transmission and, as described below, once transmission is commenced, transmission of this packet is not interrupted by other packets in the logical channel queue LC_i, even if a higher priority packet is subsequently enqueued.

Logical channel queues LC_i make their enqueued data available to a Radio Link Controller (RLC) 140. RLC 140 identifies radio available resources and feeds the enqueued data from the logical channel queues LC_i in each PQE 108_x to one or more of the available radio resources.

In the embodiment of Figure 3, the available radio resources comprise a pool 144 of dedicated data channels (DDCHs) 148_1 through 148_k and a pool 152 of shared broadcast data channels (BDCHs) 156_1 through 156_q. In system 20, DDCHs 148 and BDCHs 156 can be assigned different amounts of the total capacity of radio link 40. In a present embodiment of the invention, three BDCHs 156 are available in pool 152 and a number, typically anywhere between twenty and eighty, of DDCHs 148 can be available in pool 144 and the DDCHs 148 can have different amounts of capacity assigned to them. The creation and deletion of DDCHs 148 and the assignment and/or reassignment of DDCHs 148 is performed dynamically by system 20.

DDCHs 148 are data channels assigned to a CPE 48 for the duration of a connection session. A DDCH 148 can only have a single CPE 48 assigned to it and as a DDCH 148 can have a variable amount of capacity (effectively a data transmission rate) assigned to it, a CPE 48 only has one DDCH 148 assigned to it. The amount of capacity assigned to a particular DDCH 148 can be dynamically changed by system 20 as the requirements for the session change and/or as the amount of overall resources required or available in the sector of the base station 44 change. There is some overhead and/or delay in setting up and assigning a DDCH 148 for a CPE 48 and/or resizing or tearing down such a channel.

BDCHs 156 are broadcast channels and are available whenever the base station 44 is operating. Each CPE 48 served by a base station 44, or by a sector (a shared radio link 40) in the case of a multi-sector base station 44, receives and monitors at least one, and typically three, of the BDCHs 156 in pool 152 for data addressed to the CPE 48. Because BDCHs 156 are broadcast channels, after start up of a base station 44 there is no additional requirement or overhead to setup the BDCH channel to send data to a CPE 48. Each BDCH 156 transmits blocks of information and can transmit blocks to any CPEs 48 receiving it by addressing the block to the intended CPE 48. A particular embodiment of a suitable BDCH 156 is described in published PCT application WO 01/91407 (to Mantha), assigned to the assignee of the present invention.

Suitable methods for managing the capacity resources (i.e. – admitting CPEs 48 to the network and providing capacity to them) of a base station 44, or of the sectors of a multi-sector base station 44, will be apparent to those of skill in the art and are outside the scope of the present invention and will not be discussed further herein. However, the number of DDCHs 148 and their capacities and the scheduling of BDCH blocks to CPEs 48 are all factors that QoS processor100 and RLC 140 must cope with. In a present embodiment of the invention, the logical channel queues of a PQE 108_x can be assigned to any two of the available BDCHs 156 that the destination CPE 48 is receiving and to one of the available DDCHs 148, although as mentioned above these DDCHs 148 can have different maximum data transfer rates, etc. As will be apparent to those of skill in the art, the present invention is not limited to being assigned to two BDCHs 156, nor to one DDCH 148 and in fact, having more than one DDCH 148 assigned permits implementation of desirable features such as soft or softer hand-off between sectors or base stations wherein each downlink DDCH 148 is transmitted from a different sector or base

station 44.

RLC 140 performs the prioritization, segmentation and, if desired, traffic shaping of data packets for transmission over the available radio resources. Specifically, RLC 140 includes a segmentation cache 160 for each logical channel queue LC_i in each PQE 108_x. Each segmentation cache 160 takes the highest priority packet, at any respective time, from its respective logical channel queue LC_i and stores it within the cache until it has been completely transmitted over one or more of BDCHs 156 or DDCHs 148. Thus, the above-mentioned non-preemptive (within the logical channel queue) scheduling of the transmission of the highest priority packet in each logical channel queue LC_i is achieved.

RLC 140 communicates with each PQE 108_x and with pool 152 and pool 144 to determine the transmission requirements for the logical channel queues and the radio resources available to serve them. RLC 140 examines the segmentation caches $160_{x,i}$ for each PQE 108_x to determine the cached packet, or remaining portion of a packet, with the highest priority. Pools 144 and 152 report the available capacity of each of their respective channels assigned to a PQE 108_x to RLC 140 which will then select the contents of a segmentation cache $160_{x,I}$ for the next transmission on a channel (DDCH or BDCH) assigned to that PQE 108_x channel.

For example, in Figure 3 PQE 108_x has both BDCHs 156_1 and 156_q and DDCH 148_1 assigned to it. Specifically, LC_0 and LC_{15} have been assigned to DDCH 148_1, LC_1 has been assigned to BDCH 156_1 and LC_{15} has been assigned to BDCH 156_q. Pool 144 will report its capacity to transmit data on its next transmission frame to RLC 140. For example, pool 144 can report that DDCH 148_1 has the data transmission capacity to transmit twenty-four bytes of data on the next transmission frame. Similarly, pool 152 can report to RLC 140 that BDCHs 156_1 and 156_q have the data transmission capacity to transmit forty eight and ninety six bytes of data respectively on their next transmission frames.

RLC 140 examines this reported data transmission capacity for DDCH 148_1 and the relative priorities of the data in segmentation caches $160_{x,0}$ and $160_{x,15}$ and selects one of the two caches for transmission, for example the data in segmentation cache $160_{x,0}$. RLC 140 then segments the data in segmentation cache $160_{x,0}$, if necessary, to fit within the reported data transmission capacity of DDCH 148_1 (i.e. – twenty four bytes in the above-mentioned example). This segmented data is then provided to DDCH 148_1 for packaging and transmission in the next frame.

Similarly, RLC 140 examines the reported data transmission capacities for BDCHs 156_1 and 156_q and the contents of the segmentation caches 160 assigned to each of these BDCHs. For each BDCH 156, RLC 140 selects the assigned segmentation cache 160 whose contents have the highest priority. RLC 140 segments the data in that segmentation cache 160, if necessary, to fit within the reported data transmission capacity for the BDCH 156 and this data is then provided to the BDCH 156 for packing and transmission in the next frame.

As will be apparent to those of skill in the art, should the contents of the segmentation cache 160 with the highest priority not require all of the reported data transmission capacity of a DDCH or BDCH, RLC 140 can also include another segment, from the contents of the next highest priority segmentation cache 160 or from another packet from the same logical channel queue which is loaded into the highest priority segmentation cache 160, to utilize all of the available reported data transmission capacity. This can occur, for example, when the highest priority cache contains the last portion of a packet to be transmitted or contains a very small packet, and the data to be transmitted is less than the reported data transmission capacity. If the channel under consideration is a BDCH 156, RLC 140 can also add another segment from a logical channel queue assigned to the BDCH 156 from another PQE 108.

Figure 5 shows a flowchart of the above-described process. As indicated at step 200, the method considers, in turn, each and every channel to which one or more segmentation caches 160 are assigned. At step 204, the non-empty segmentation cache 160 of the logical channel queue with the highest priority that is assigned to the channel under consideration is selected. At step 208, a segment of the contents of the selected segmentation cache 160 is formed, if necessary, to fit the data transmission capacity reported to RLC 140 by the channel. At step 212, the segment is presented to the channel for packaging and transmission. At step 216, a check is made to see if all of the reported data transmission capacity has been used. If all the capacity has been used, the process returns to step 200 for the next channel. If less than the total amount of reported data transmission capacity has been used, the process returns to step 204 where, if the channel under consideration is a BDCH 156, all of the logical channel queues (from any PQE 108) assigned to the BDCH 156 are considered by RLC 140 to be included as an additional segment. If the channel under consideration is a DDCH 148, RLC 140 will consider the next highest priority data from the same PQE 108 to be included as an additional segment.

This process is performed by RLC 140 for each frame transmitted from base station 44,

which can result in the contents of a different segmentation cache 160 being selected for the next transmission before the contents of another segmentation cache are completely transmitted. Thus, RLC 140 implements a preemptive scheduling method between logical channel queues and/or between PQE's 108.

Figure 6 shows an example of the results of preemptive scheduling in progress for one DDCH 148 that has at least logical channels LC_0, LC_2 and LC_3 of a PQE 108 assigned to it. As shown in the Figure, in the frame transmitted commencing at time t_0, the data in the segmentation cache for logical channel queue LC_3 is segmented to the reported transmission capacity for DDCH 148 and is transmitted. Another segment of this data is formed and transmitted in the frame commencing transmission at time t_1. While the illustration in the Figure indicates that the segments are the same size, this need not be the case and the reported data transmission capacity can change frame to frame.

For the frame transmitted commencing at time t_2, further transmission of the contents of logical channel queue LC_3 are preempted by the contents that have been placed into the segmentation cache for logical channel queue LC_2 which RLC 140 determines have a higher priority. Thus, in the frame commencing transmission at time t_2, a segment of the contents of the segmentation cache 160 logical channel queue LC_2 is formed for the reported data transmission capacity and is transmitted. For the frame commencing transmission at time t_3, RLC 140 determines that the contents that have been placed in the segmentation cache 160 for logical channel queue LC_0 have yet a higher priority. Accordingly, transmission of the contents of LC_2 is preempted and a segment of logical channel queue LC_0 is formed for the reported data transmission capacity and is transmitted in the frame at time t_3. As no higher priority data is found in the segmentation caches 160 for the logical channel queues assigned to the DDCH 148 during the time periods t_4 and t_5, segmentation and transmission of the remaining contents of the segmentation cache 160 for channel queue LC_0 occur at times t_4 and t_5 until all three segments, representing the entire packet in the segmentation cache for logical channel queue LC_0 have been transmitted.

At time t_6, RLC 140 determines the segmentation cache 160 with the highest remaining priority for transmission. In the illustrated example, the next highest priority data is the remaining portion of the data in the segmentation cache for logical channel queue LC_2 and this data is segmented to fit the reported data transmission capacity and is transmitted in the frames

commencing transmission at time t_6 and t_7.

At time t_8, RLC 140 determines the segmentation cache with the highest remaining priority for transmission. In the illustrated example, the next highest priority data is the remaining data in the segmentation cache 160 for logical channel queue LC_3 and this data is segmented to fit the reported transmission capacity and is transmitted in the frames commencing transmission at times t_8, t_9, t_{10} and t_{11} at which time the complete packet has been transmitted.

At each CPE 48, a reassembly controller 240 is provided, as illustrated in Figure 7. Reassembly controller 240 receives data from each BDCH 156 and DDCH 148 channel that are transmitted to it over radio link 40. Data received by reassembly controller 240 is first examined by sequencer 244 that determines which, if any, existing logical channel queue LC at base station 44 was the source of the data. Sequencer 244 will place the received data into the reassembly queue RQ_i associated with the identified logical channel queue LC. If sequencer 244 determines that the data was transmitted by a newly created logical channel queue LC at base station 44, sequencer 244 will cause a complementary reassembly queue RQ_i to be instantiated in reassembly controller 240 and will place the received data therein.

Each reassembly queue RQ_i examines data placed into it and reassembles the data segments into the original data packet. Once reassembly of a packet has been completed, it is output to the appropriate data target in CPE 48 and the reassembly queue RQ_i is emptied for the next received data.

If transmission of a packet from base station 44 did not require segmentation, the received data packet is placed into reassembly queue RQ_i that then immediately outputs it.

As will be apparent to those of skill in the art, a variety of techniques can be employed for dealing with dropped/lost packet segments and received erroneous packets and these techniques are outside the scope of the present invention and will not be described herein.

Figure 8 shows the prioritization and queuing engine PQE 300 which runs at each CPE 48 in system 20. PQE 300 is similar to PQE 108, although it is somewhat simpler as each CPE 48 only has access to a single uplink DDCH channel 148 on radio link 40 to base station 44. An uplink DDCH channel 148 is assigned to a CPE 48 when needed and, as mentioned above, can provide different amounts of data transmission capacity to meet the needs of the CPE 48 to which it is assigned. As was mentioned above, suitable methods for managing the capacity resources (i.e. – admitting CPEs 48 to the network and providing uplink capacity to them) in

system 20 will be apparent to those of skill in the art and are outside the scope of the present invention and will not be discussed further herein.

PQE 300 includes a network interface port 304 to which applications and/or processes running on the CPE 48, or the data terminals attached to it, send the information to be transmitted to base station 44. Data received at port 304 is classified by classifier 308, which performs the same functions as classifier 112, discussed above with reference to PQEs 108. In the illustrated embodiment of the invention, each PQE 300 includes sixteen logical channel queues, specifically, Lc_0 through Lc_{15} and each logical channel queue Lc_i has an appropriate priority and set of QoS attributes defined for it. While it is contemplated that one or more logical channel queues will be predefined for PQE 300, additional logical channel queues Lc_i can be created or removed, as desired, and the QoS attributes defined for each logical channel queue Lc_i can be set and reset as required.

Each logical channel queue Lc_i has an associated segmentation cache 312_i defined for it that performs the same functions as segmentation caches 160, discussed above with reference to PQEs 108.

Each CPE 48 also includes a radio link controller RLC 316 that operates to receive reported data traffic capacity information from DDCH 148 and to select the segmentation cache 312_i whose contents have the highest priority for transmission. If necessary, PLC 316 will segment the contents of the selected cache 312_i to fit the reported data traffic capacity and will provide the data (whether segmented or not) to DDCH 148 for transmission as the next transmitted frame.

At base station 44, a reassembly controller (not shown) is provided for each uplink DDCH 148 then assigned to a CPE 48. These reassembly controllers are similar to those discussed above with respect to Figure 7, and operate to reassemble segmented packets transmitted over the DDCHs 148 before forwarding the packets to their destination in base station 44 or elsewhere.

As will now be apparent, the present invention provides QoS services for data flows with wildly different requirements over a link that has time varying capacities. The multiple logical channels queues share a single link and yet each queue can be provided with different QoS attributes appropriate to its needs even while the data transmission capacity of the link changes with time. In the downlink direction, the link can be structured into multiple channels and the

logical channel queues can be mapped to one or more of these channels. These channels can include dedicated channels, each dedicate to a link between the base station 44 and a CPE 48, with variable data transmission capacities and one or more channels can be broadcast channels from the base station 44 to several, or all, of CPEs 48. In the uplink, the logical channel queues are mapped to a single dedicated channel that can have a variable data transmission capacity.

One of the advantages of the present invention is that it allows QoS attributes to be defined and provided on a per logical channel basis which allows for network resources to be used efficiently, to provide differentiated QoS on a per data flow basis and to support QoS services over a wireless link. This allows, for example, the logical channel for a media connection such as a voice telephony data flow, to be defined with attributes for segmentation prohibition, low latency and low reliability which are suitable for such a connections, as a voice call is sensitive to latency, but can tolerate some dropped packets and its packets are always of a known size, as required by the particular codec employed.

In such a situation, RLC 140 will attempt to ensure that packets in such a logical channel queue are transmitted with the required time periods (to meet the latency requirements) but without ARQ or other reliability techniques being applied to it. Conversely, a file transfer, such as an ftp session, between a CPE 48 and a base station 44 can be transmitted through a logical channel queue that has defined attributes for being latency tolerant, but requiring high reliability. Similarly, fax data may be transmitted through a logical channel queue that has defined attributes for being latency intolerant and requiring high reliability, so that ARQ or other reliability techniques are applied. As mentioned, reliability techniques such as ARQ can be provided on a per logical channel basis. Further, less conventional attributes, such as higher error correcting coding levels or even radio transmission power level margins can also be defined and implemented on a per logical channel basis. Also, other attributes such as whether and which types of header compression to apply to a flow can be defined per flow.

Further, traffic shapers can be implemented and configured on a per logical channel basis. This allows, for example, voice telephony data to be transferred over link 40 as necessary, while other data types can be data rate limited according to parameters defined by the network operator. Thus, a telephony call can be conducted unimpeded while a file transfer or other large data transfer can be subject to a leaky bucket, or other traffic shaping process.

As should now be apparent to those of skill in the art, the unique flexibility described

above is achieved with a very low overhead impact on the transmission link capacity. Transmission of segments involves only the addition of a small header to each segment, the header identifying the sending logical channel and, in the case were segmentation of the packet has occurred, an indication of the segmentation of the packet and the location of the segment within the full packet.

The present invention provides for the efficient utilization of a shared resource, such as a radio link, in a network including heterogeneous links. Data is arranged in flows and each flow can be provided with its own logical channel with its own set of QoS attributes. These attributes can include prioritization, latency restrictions, data rate requirements, reliability requirements, etc.

The above-described embodiments of the invention are intended to be examples of the present invention and alterations and modifications may be effected thereto, by those of skill in the art, without departing from the scope of the invention which is defined solely by the claims appended hereto.